

REMARKS

[0002] Applicant respectfully requests entry of the following remarks and reconsideration of the subject application. Applicant respectfully requests entry of the amendments herein. The remarks and amendments should be entered under 37 C.F.R. §1.116 as they place the application in better form for appeal, or for resolution on the merits.

[0003] Applicant respectfully requests reconsideration and allowance of all of the claims of the application. Claims 1-11 and 13-22 are presently pending. Claims amended herein are 1-4, 7-11, 13-16, and 18-22. Claim 12 is canceled without prejudice or disclaimer.

Formal Request for an Interview

[0004] If the Examiner's reply to this communication is anything other than allowance of all pending claims, then I formally request an interview with the Examiner. I encourage the Examiner to call me—the undersigned representative for the Applicant—so that we can talk about this matter so as to resolve any outstanding issues quickly and efficiently over the phone.

[0005] Please contact me or my assistant to schedule a date and time for a telephone interview that is most convenient for both of us. While email works great for us, I welcome your call to either of us as well. Our contact information may be found on the last page of this response.

Claim Amendments

[0006] Without conceding the propriety of the rejections herein and in the interest of expediting prosecution, Applicant amends claims 1-4, 7-11, 13-16, and 18-22 herein.

[0007] The term “component” in various claims has been replaced with “handler component” for clarification purpose. The support for the amendment can be found throughout the Application including, for example, Fig. 2 (Extensible Configuration Handler 152 and Non-Extensible Configuration Handler 160) with the associated text.

[0008] Claim 1 is amended to recite computer readable media having stored thereon a plurality of instructions that cause the one or more processors to:

“access a configuration file written in a markup language and associated with an application, the configuration file having definitions of a plurality of configuration handlers for creating handler components based on a mapping table defined in the configuration file, at least one of the plurality of configuration handler definitions including a definition of a first configuration handler, the first configuration handler being configured to create a first handler component based on the mapping table, and further comprising a second configuration handler nested within the first configuration handler, the second configuration handler being configured to create a second handler component based on a mapping definition in a configuration section within the first configuration handler...”

[0009] Support for the various amendments can be found throughout the Application including, for example, Tables I and II with associated text. With respect to

“configuration file written in a markup language,” it is disclosed in the original claim 2 that “each of the plurality of component definitions being written in an extensible Markup Language (XML) format.” It’s also disclosed in the Specification that “[t]he configurable data definitions may be written using a markup language.” (Specification at p.6, lines 14-15). Furthermore, Tables I and II, which include examples of a portion of a configuration file, are also written in a markup language. (See Tables I and II in Specification at pp. 12 and 24).

[0010] Furthermore, claim 1 is amended to recite, *inter alia*, “wherein the second configuration handler is user-defined to handle a customized handler data that is not supported by the first configuration handler...” Support for the amendment can be found throughout the Application including, for example, description in Specification wherein “[t]he routerConfigHandler configuration handler was not written to understand the routing definition format of the myRoute tag. So, when the routerConfigHandler configuration handler encounters the tag myRoute in line 13, the routerConfigHandler configuration handler refers to the mappings of lines 7-9 and passes the content associated with the myRoute tag (e.g., the content of lines 13-15) to the Router.myRouteConfigHandler child configuration handler, creating the Router.myRouteConfigHandler child configuration handler if necessary.” (Specification at p.14, lines 2-9 in view of Table I).

[0011] Moreover, claim 1 is amended to recite, *inter alia*, “wherein the second handler component created by the second configuration handler implements a known interface such that the data defined by the second configuration handler is properly processed by the application without alteration to the application...” Support for the amendment can be found throughout the Application, including description wherein

“It should be noted that, although such extensible configuration handlers are extensible, they can still implement a known interface. The data format(s) that the extensible configuration handlers define for their values are extensible, as is the code that is used to read such data, but these extensions are still tied to this known interface. Thus, other components or applications need not be altered to make use of components created by such extensible configuration handlers, and configuration system 106 need not be altered to make use of the extensible configuration handlers.”

(Specification at p.7, lines 14-21).

[0012] Claims 7, 11, 14, and 21 are similarly amended, and therefore are supported by the Application as well.

[0013] Accordingly, no new matter will be introduced by this paper. Entry to the file is respectfully requested.

Substantive Matters

Claim Rejections under § 103

[0014] Claims 1-5, 7, 9-10 stand rejected under 35 U.S.C. § 103 for being unpatentable over U.S. Patent No. 7,117,504 to Smith et al. (“Smith”) in view of U.S. Patent Application Publication No. 2005/0055680 to Kluger et al. (“Kluger”).

[0015] Claims 6, 8, 11-22 stand rejected under 35 U.S.C. § 103 for being unpatentable over Smith in view of U.S. Patent Application Publication No. 2003/0225870 to Sandadi et al. (“Sandadi”).

[0016] In light of the amendments presented herein, Applicant submits that these rejections are moot. Accordingly, Applicant asks the Examiner to withdraw these rejections.

[0017] Independent claim 1, as amended, recites:

1. One or more computer readable media having stored thereon a plurality of instructions that, when executed by one or more processors, causes the one or more processors to:

access a configuration file written in a markup language and associated with an application, the configuration file having definitions of a plurality of configuration handlers for creating handler components based on a mapping table defined in the configuration file, at least one of the plurality of configuration handler definitions including a definition of a first configuration handler, the first configuration handler being configured to create a first handler component based on the mapping table, and further comprising a second configuration handler nested within the first

configuration handler, the second configuration handler being configured to create a second handler component based on a mapping definition in a configuration section within the first configuration handler, wherein the second configuration handler is user-defined to handle a customized handler data that is not supported by the first configuration handler, and wherein the second handler component created by the second configuration handler implements a known interface such that the data defined by the second configuration handler is properly processed by the application without alteration to the application;

create the plurality of handler components including the first handler component and the second handler component in accordance with the definitions;

inform one or more of the plurality of handler components of the presence of other handler components; and

make the plurality of handler components available to the application.

[0018] In making out the rejection to claim 1, the Office took the position that “Smith discloses access a configuration file associated with an application, the configuration file have a plurality of component definitions...; create a plurality of components, each component being created based on one of the plurality of component; inform one or more of the plurality of components of the presence of other components...; make the plurality of components available to the application...” (Office Action 08/27/2007 at p.2-3).

[0019] Applicant respectfully submits that the rejections are moot in light of the amendment. In particular, more contexts have been recited in the amended claim 1 to distinguish over Smith.

[0020] Smith is directed to network software, such as Web application, and to computer software development of such network software application program interface (API) that facilitates use of a network software platform. Smith discloses a developers' programming framework in Fig. 2, and outlines of namespaces in Fig. 3. According to Smith, "the base class library namespace 206 ("System") includes... a configuration namespace 362 ("System.Configuration") containing classes and interfaces that allow developers to programmatically access configuration settings and handle errors in configuration files... " However, that particular configuration namespace does not support the Office's position that Smith discloses "the configuration file have a plurality of component definitions". Moreover, claim 1 is amended to recite that "the configuration file having definitions of a plurality of configuration handlers for creating handler components based on a mapping table defined in the configuration file..." This feature is not disclosed or suggested in the configuration namespace either.

[0021] Furthermore, Applicant respectfully submits that "creating the plurality of handler components including the first handler component and the second handler component in accordance with the definitions" as recited in amended claim 1 is missing in Smith. In rejecting claim 1, the Office referred to Smith ("col. 3, ln. 63-67, services created across system & col. 4, ln. 10-14 & ln. 25-27 and can be combined with each other or defined by each other"). (Office Action 08/27/2007 at p.2). Applicant respectfully disagrees.

[0022] The "Web services" as described in Smith, "can be accessed over a network... The Web services... are programmable application components that are reusable and interact programmatically over the network, typically through industry standard Web protocols, such as XML, SOAP, WAP, HTTP, an SMTP although other

means of interacting with the Web services over the network may also be used...” (Smith, col. 3, line 65-col. 4, line 8). Smith, however, does not specifically disclose that the Web services are created “in accordance with the definitions” in a configuration file.

[0023] Moreover, it’s respectfully asserted that the “handler component” recited in claim 1 is not “Web services” defined in Smith. In particular, the feature that handler component is configured to “handle ...handler data” is completely missing in Smith. The “Web services” are defined in Smith as “application components” to perform a variety of different services including login verification, notification, database storage, stock quoting, location directories, mapping, music, electronic wallet, calendar/scheduler... and so on. (Smith, col. 4, lines 2-25). Nonetheless, Smith does not specifically disclose that a Web service is created to handle handler data.

[0024] While admitting that Smith does not explicitly teach the “nest structure” recited in claim 1, the Office brought Kluger to remedy the deficiency. In particular, the Office noted that “Kluger teaches multiple definitions and nested handling for data objects (Par. 24, ln 9-15).” (Office Action 08/27/2007 at p.3). Applicant respectfully disagrees.

[0025] Kluger is directed to method and apparatus for defining and using data types at runtime. According to Kluger, a procedural source code program is compiled to create a runtime data type and to use the runtime data type upon a type checking. (Kluger, Abstract). Kluger further discloses that “[f]or the virtual machine to be able to do a dynamic data check at runtime...each data object used in a program includes a reference to its own data type. Other information is also available for improving the performance of operations on data objects, for example, an initial value for rapid

initialization of data objects, views of (nested) structures for efficiently structure handling, and a hash value of the technical properties of the type for rapid compatibility testing.” (Kluger, paragraph [0024]).

[0026] Applicant respectfully submits that reference Kluger does not disclose the nest structure as recited in amended claim 1. In particular, the above-quoted description from Kluger does not support Office’s position. For example, “views of (nested) structures for efficiently structure handling” in Kluger does not suggest or disclose that “the first configuration handler [being] configured to create a first handler component based on the mapping table, and further [comprising] a second configuration handler nested within the first configuration handler, the second configuration handler being configured to create a second handler component based on a mapping definition in a configuration section within the first configuration handler...”, because the runtime data object in Kluger does not comprise a “second handler component” created by a second configuration handler based on “a mapping definition in a configuration section within the first configuration handler.” Instead, the runtime data object, which is “defined from the compound structure of data types that are known at compile-time or previously-defined runtime data types”, is created at runtime as one single data object. (Kluger, paragraph [0007]).

[0027] Furthermore, the feature that “the second configuration handler being configured to create a second handler component based on a mapping definition in a configuration section within the first configuration handler...” is missing in Kluger. As disclosed in Kluger, a runtime data object is defined from a compound structure of data types. According to Kluger, “the class CL_ABAP_DATATYPE 310 is the most general class in the class hierarchy 300 and provides an interface to types that can be generally

used to define runtime data types...class CL_ABAP_STRUCTURE, CL_ABAP_TABLE, and CL_ABAP_REFERENCE [inherit] from CL_ABAP_DATATYPE...” (See Fig. 3 and associated paragraphs [0027]-[0033]). As known in the related art, the inherent structure in Kluger is different from the configuration structure recited in claim 1. In inherent structure, a new data object is defined by inheriting attributes and methods from known data type, rather than by comprising a child component in the known data type. Therefore, the structure recited in claim 1 is not disclosed or suggested in Kluger.

[0028] Moreover, the amended features “the second configuration handler is user-defined to handle a customized handler data that is not supported by the first configuration handler” as well as “the second handler component created by the second configuration handler implements a known interface such that the data defined by the second configuration handler is properly processed by the application without alteration to the application...” is completely missing in Smith or Kluger.

[0029] Therefore, claim 1 is asserted patentable over Smith and Kluger. Claim 7 incorporates the similar feature, and therefore is asserted patentable over Smith and Kluger too.

[0030] Independent claim 11, as amended, recites:

11. One or more computer readable media having stored thereon a plurality of instructions that, when executed by one or more processors, causes the one or more processors to access a configuration file and create a plurality of handler components including at least a first and a second handler component based on a first and a second configuration handler defined in the configuration file associated with an application, the first

configuration handler comprising the second configuration handler and mapping the second configuration handler in a configuration section within the first configuration handler, wherein the second configuration handler is user-defined to handle a customized handler data that is not supported by the first configuration handler, and wherein the second handler component created by the second configuration handler implements a known interface such that the data defined by the second configuration handler is properly processed by the application without alteration to the application.

[0031] Applicant respectfully disagrees with the Office's position in paragraph [21] of Office Action dated August 27, 2007, the reason of which has been addressed in this paper. (See paragraphs [0021]-[0023] of this paper). While admitting that Smith does not explicitly disclose the nested configuration handlers, the Office brought Sandadi to remedy the deficiency ("Sandadi discloses the nested configuration handlers being used to create a plurality of components. (Par. 3 & #512, Fig. 5&6, after creating a component using nested configuration handlers to create a plurality of components") (Office Action 08/27/2007 at p.7, paragraph [22]). Applicant respectfully submits that the rejections to claim 11 are moot in light of the amendment.

[0032] Sandadi is directed to method and function component for carrying out management of client/server processes operating within separate process spaces within a computer system (Sandadi, Abstract). Sandadi provides an exemplary outline for a blocking event processing thread utilized to respond to a set of events associated with the content of the event list structure. (See Fig. 6 with the associated text).

[0033] However, the feature "[accessing] a configuration file and [creating] a plurality of handler components including at least a first and a second handler component

based on a first and a second configuration handler defined in the configuration file associated with an application” is missing in Sandadi. The components in Sandadi are not defined in a configuration file. Instead, they are defined in source code level and compiled to executable .EXE files. According to Sandadi, “[i]n a component-based software approach, an application executable comprises a set of building blocks, or components, that are defined by an interface and a set of functions (e.g., methods) rendered by the component via the interface. In the case of COM/COM+ components, the COM object components (including methods and /or data) are compiled into binary executable modules. The components, depending upon their mode of deployment (i.e., in process or out-of-process) are deployed as either .DLL file (in-process) or .EXE files (out-of-process). The present invention concerns the out-of-process form of software component.” (Sandadi, paragraph [0002]).

[0034] Applicant further submits that the feature wherein “the first configuration handler comprising the second configuration handler and mapping the second configuration handler in a configuration section within the first configuration handler” is not disclosed or suggested in Sandadi either. In particular, the pseudo code in Fig. 6 does not show that “a first configuration handler [comprising] a second configuration handler”. As disclosed in Sandadi, “[t]he thread structure includes a switch 604 (e.g., case instruction) that executes one of multiple different sets of instructions based upon the type of event that caused the thread to emerge from the wait state.” (Sandadi, paragraph [0058], lines 2-5). As known in the art, a switch statement is defined across many individual statements. A typical syntax is that the first line contains the actual word “switch” followed by either the name of a variable or some other expression allowed by the language's syntax. This variable or expression is usually referred to as the “control

variable" of the switch statement. After this line, following lines define one or more blocks of code that represent possible branches that program execution may take. (Wikipedia, http://en.wikipedia.org/wiki/Switch_statement). Accordingly, each pseudo code starting with a "case" is considered an individual handling statement/operation, and shall not be interpreted as comprising or being comprised by another parallel switch statement.

[0035] Moreover, in terms of the "switch" statement as disclosed in Fig. 6, Applicant submits that the feature "wherein the second configuration handler is user-defined to handle a customized handler data that is not supported by the first configuration handler" is not disclosed or suggested in Sandadi. As known in the art, a situation that is not supported by particular control variable of a switch statement will not be processed by that switch statement, even though the particular switch statement may contain child switch statement. Therefore, it's logically implausible for a child switch statement, if any, to handle a particular user-defined situation that is not supported by its parent switch statement.

[0036] Furthermore, Applicant submits that the pseudo code in Fig. 6 is not "defined in a configuration file associated with an application" as recited in claim 11. Rather, the pseudo code represents an outline of an algorithm in source-code level, which is actually implemented by one or more programming languages and compiled to executable .EXE files.

[0037] Accordingly, claim 11 is respectfully asserted patentable over Smith and Sandadi. Independent claims 14 and 21 incorporate at least the similar features, and therefore are asserted patentable over Smith and Sandadi too.

Dependent Claims

[0038] In addition to its own merits, each dependent claim is allowable for the same reasons that its base claim is allowable. Applicant requests that the Examiner withdraw the rejection of each dependent claim where its base claim is allowable.

Conclusion

[0039] All pending claims are in condition for allowance. Applicant respectfully requests reconsideration and prompt issuance of the application. If any issues remain that prevent issuance of this application, the **Examiner is urged to contact me before issuing a subsequent Action.** Please call/email me or my assistant at your convenience.

Respectfully Submitted,

Dated: 2008-01-28

By: Ningning Xu
Ningning Xu
Reg. No. L0293
(509) 324-9256 x226
ningning@leehayes.com
www.leehayes.com

My Assistant: Carly Bokarica
(509) 324-9256 x264
carly@leehayes.com